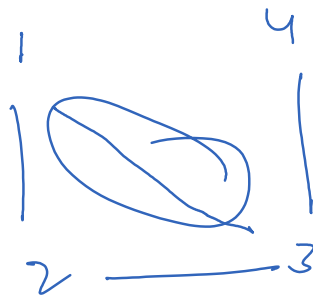
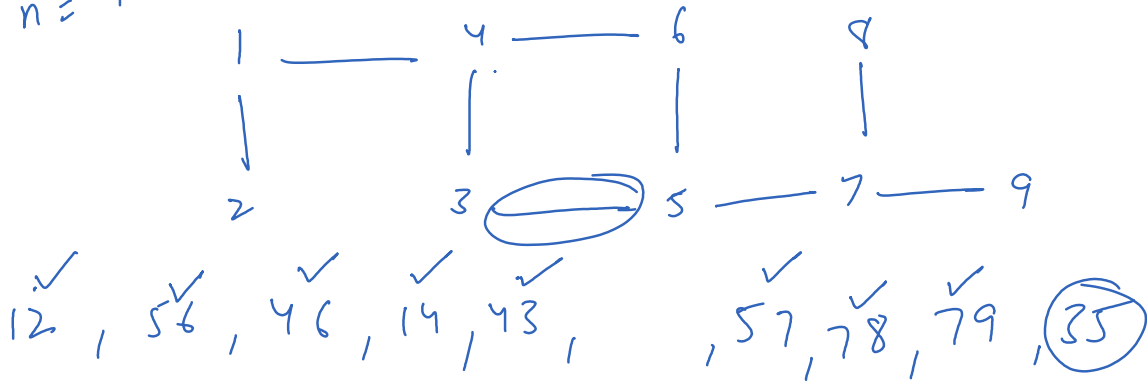
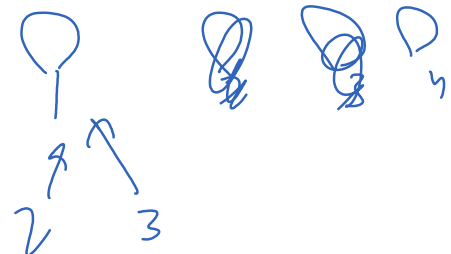


9:20 - 9:30

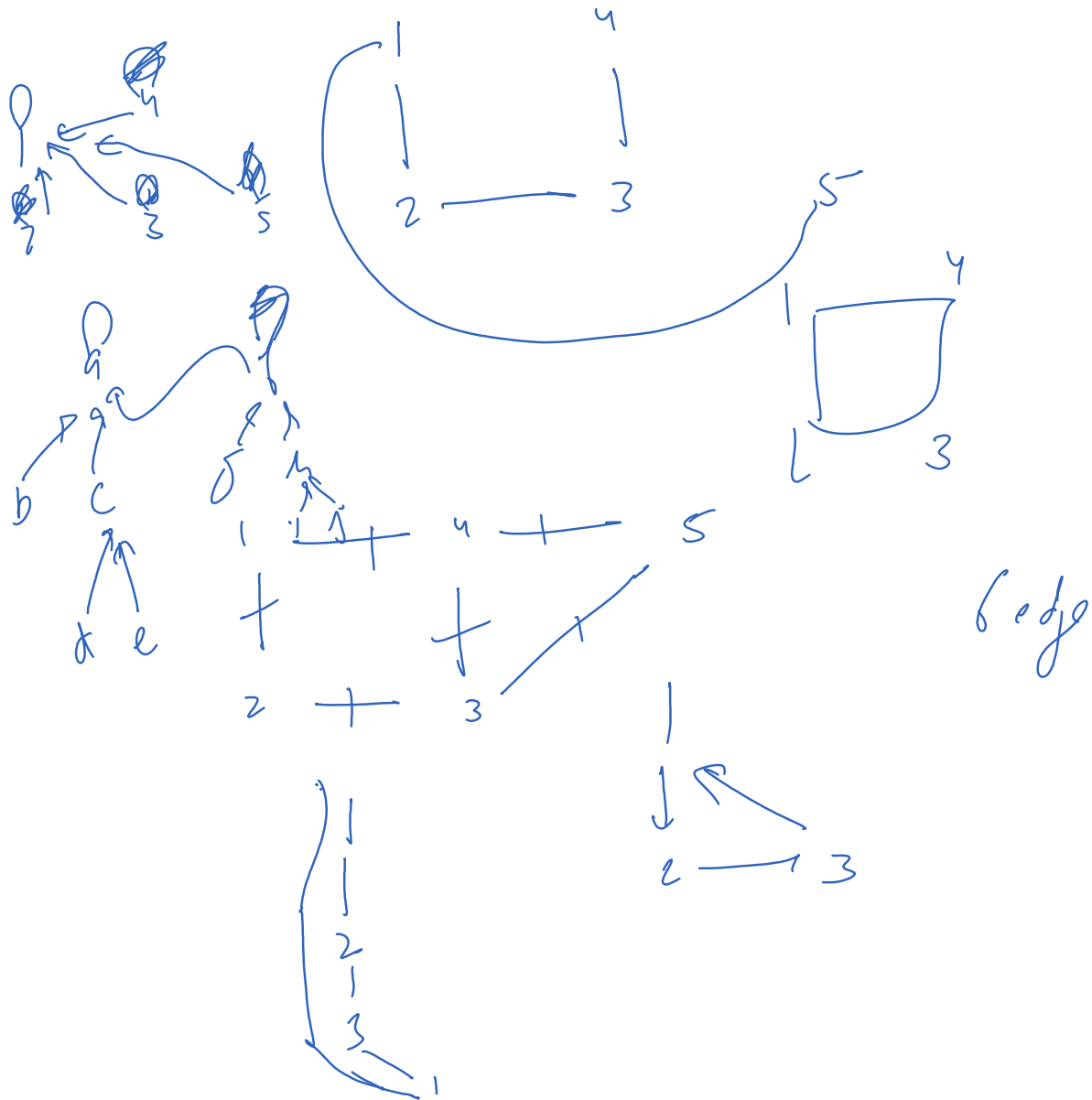
$n = 9$

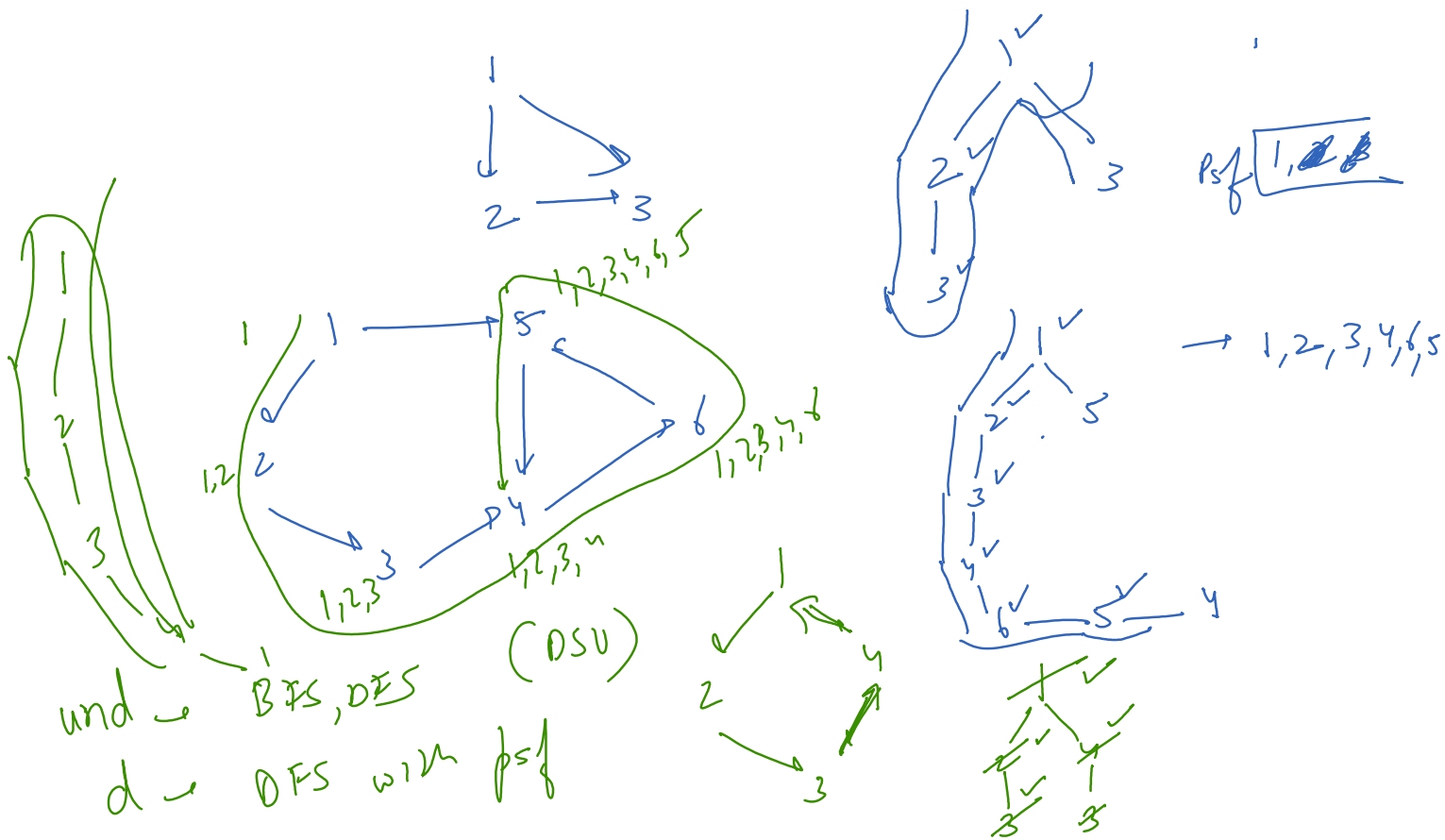


✓ 12, ✓ 23, (13), 34

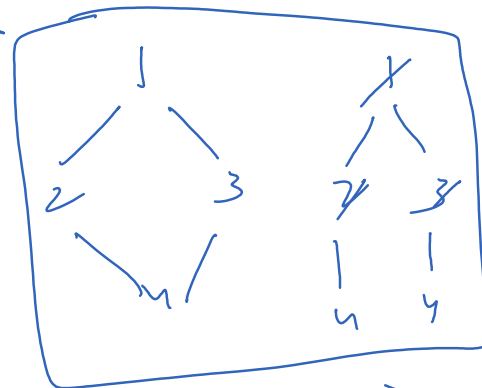


Input: edges = $[[1,2],[2,3],[3,4],[1,4],[1,5]]$
 Output: $[1,4]$



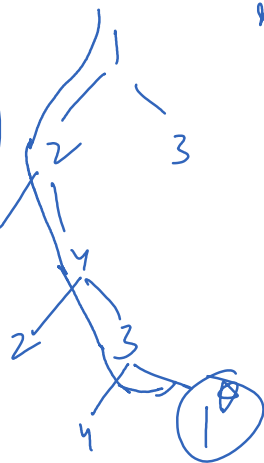


Cycle
und → BFS, DFS, DSU

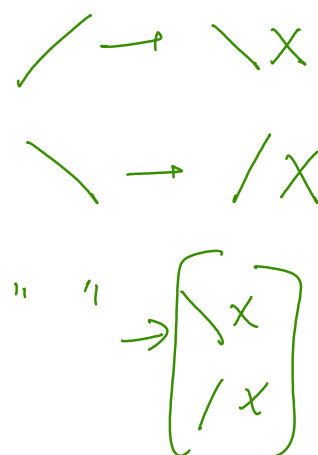
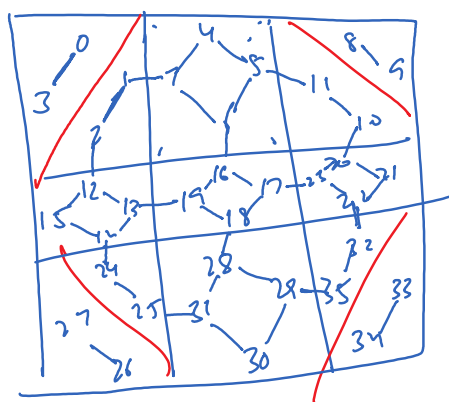
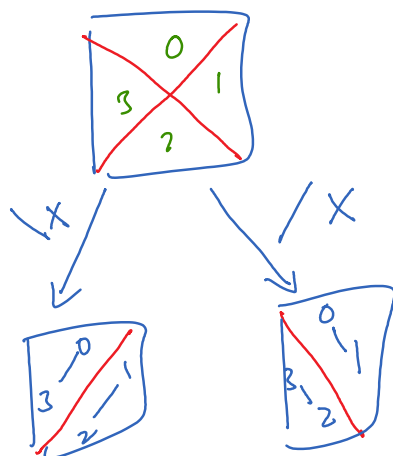
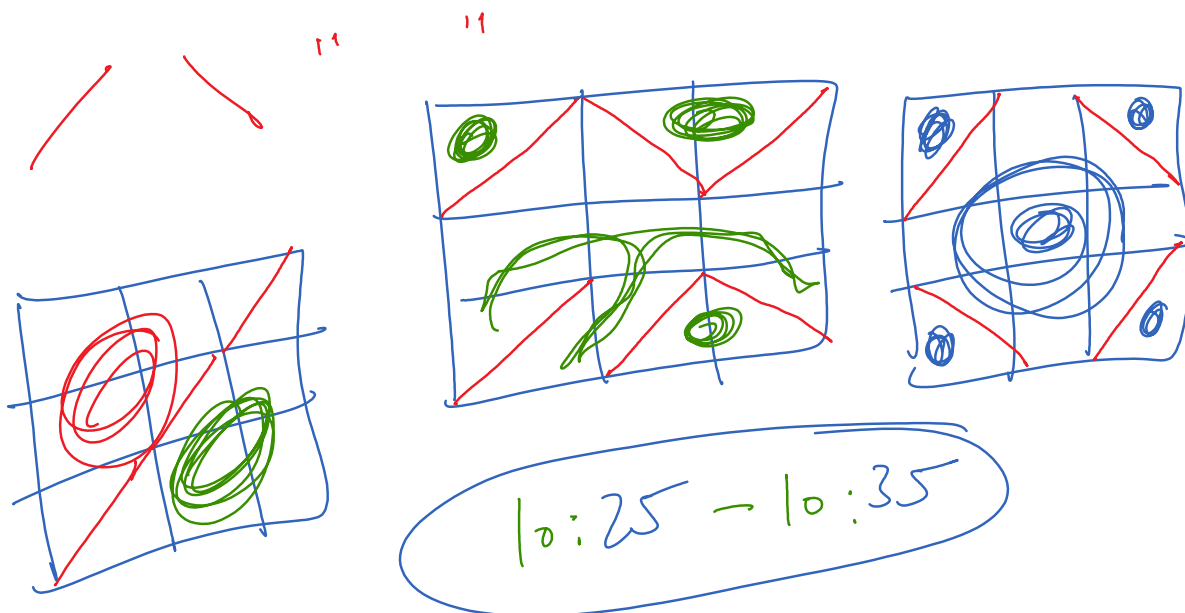


Directed → DFS with psf

DSU in directed
KANN'S



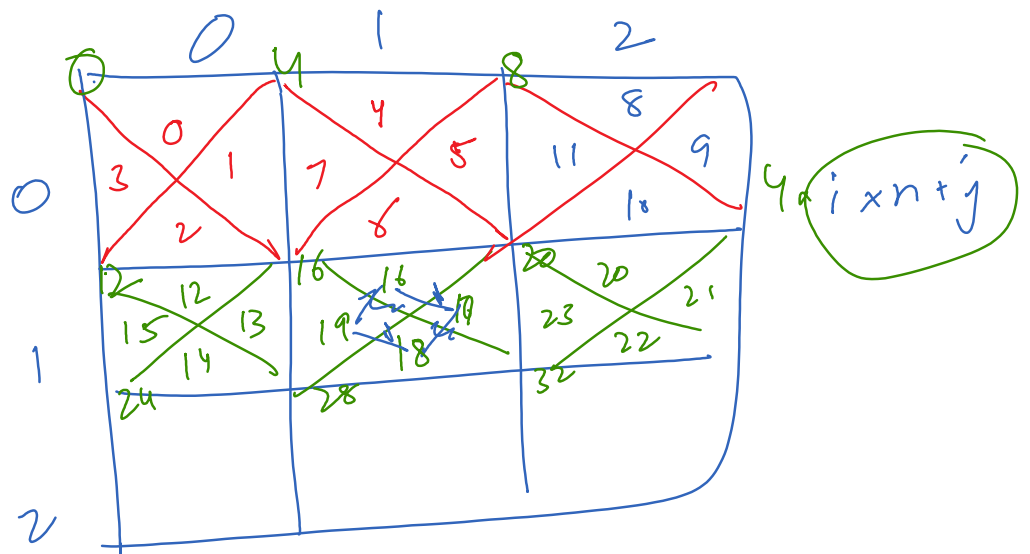
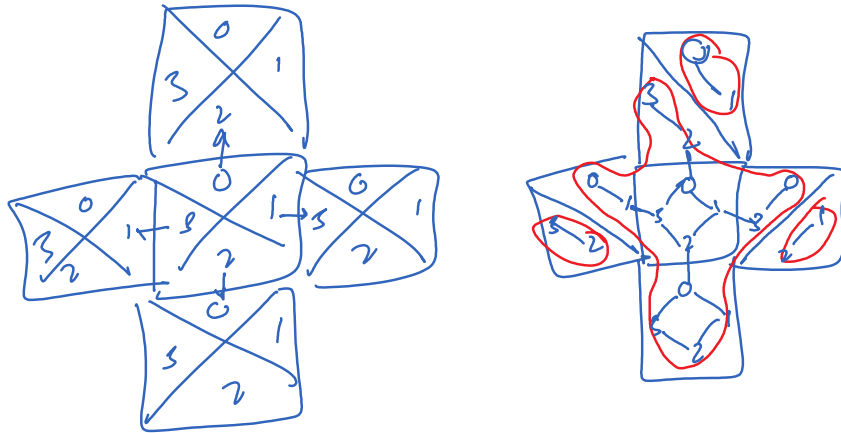
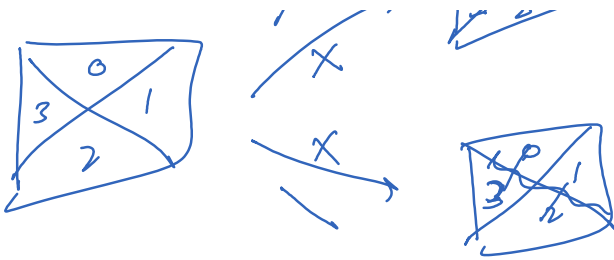
KANN'S

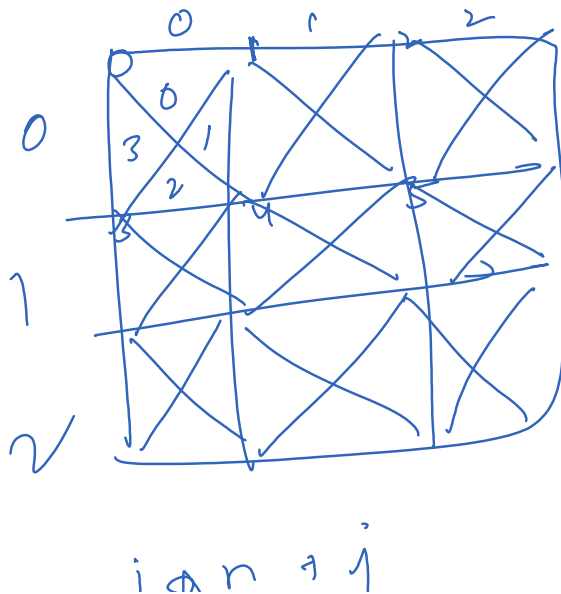
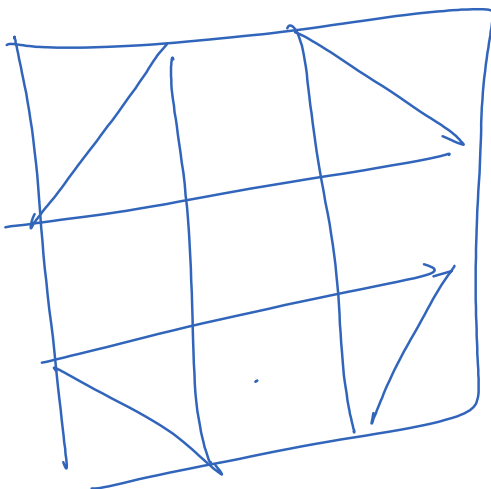
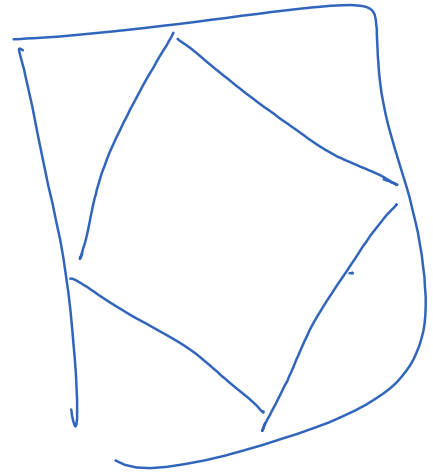
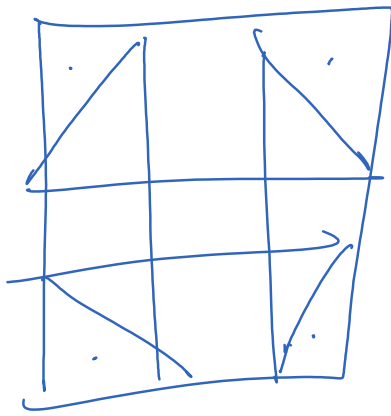
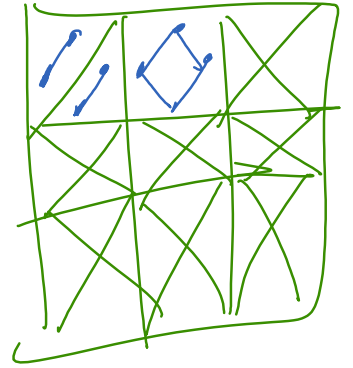
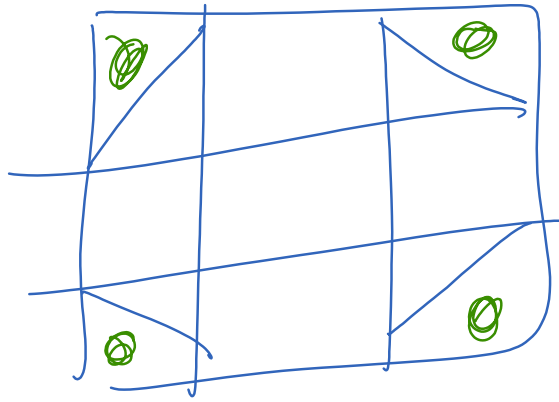


$$N \times N$$

$$3 \times 3 \times 4 = 36$$



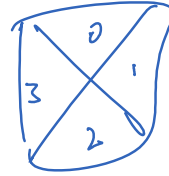




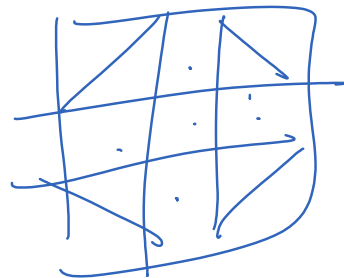
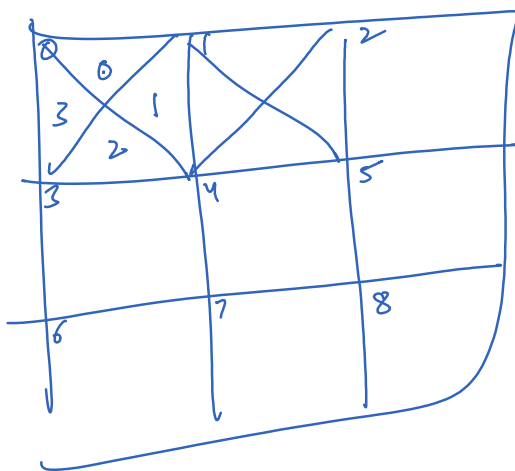
1

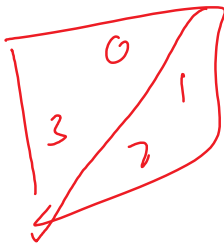
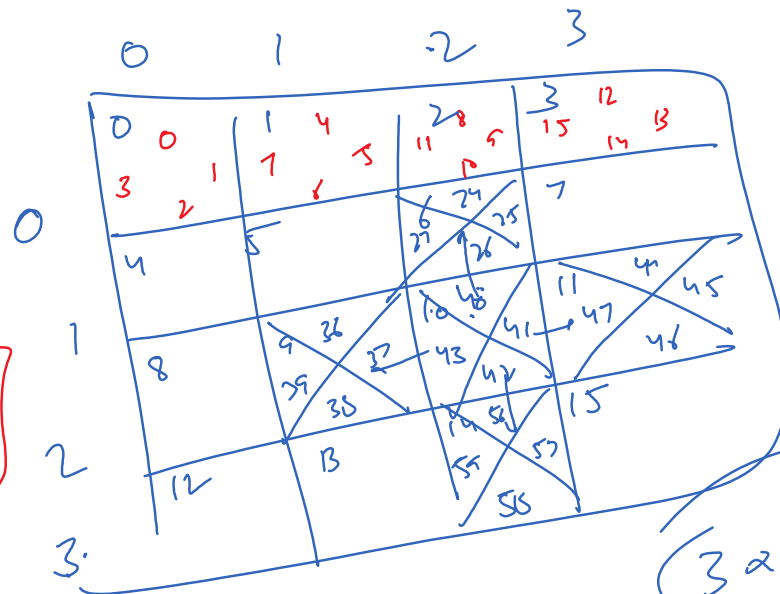
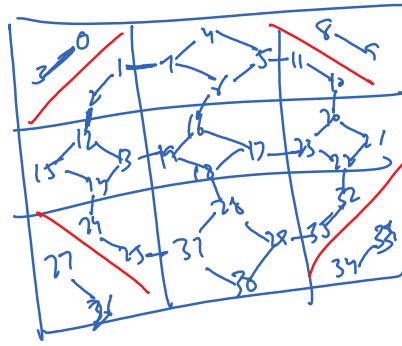
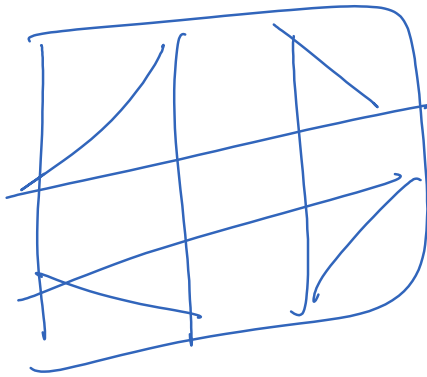
$i \oplus n + j$

	0	1	2	3
0	0	1	2	3
1	3	2	5	6
2	4	5	6	7
3	8	9	10	11



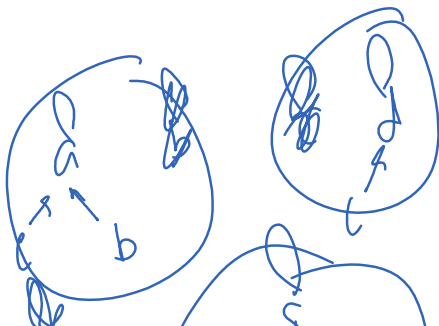
$$2 \times (4 + 1) + 0$$



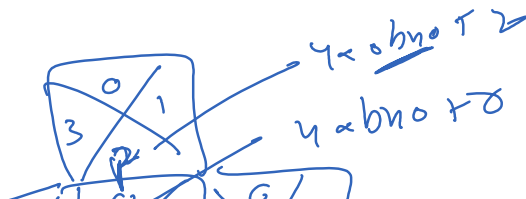


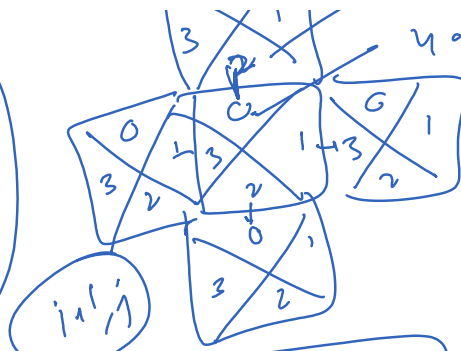
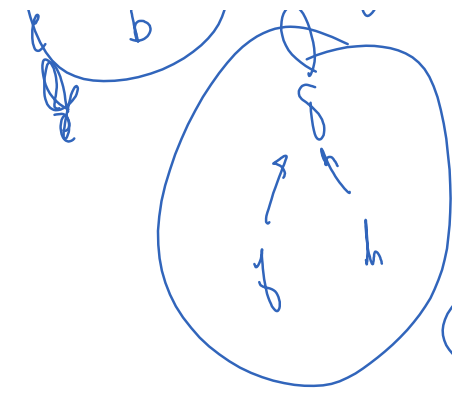
~~AT~~
T.
R L
P

$$3 \times (4) + 2$$



3, 2

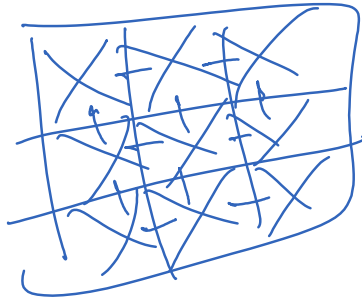




obno

$$bno = i \times n + j$$

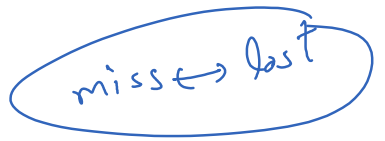
$$obno = (i-1) \times n + j$$



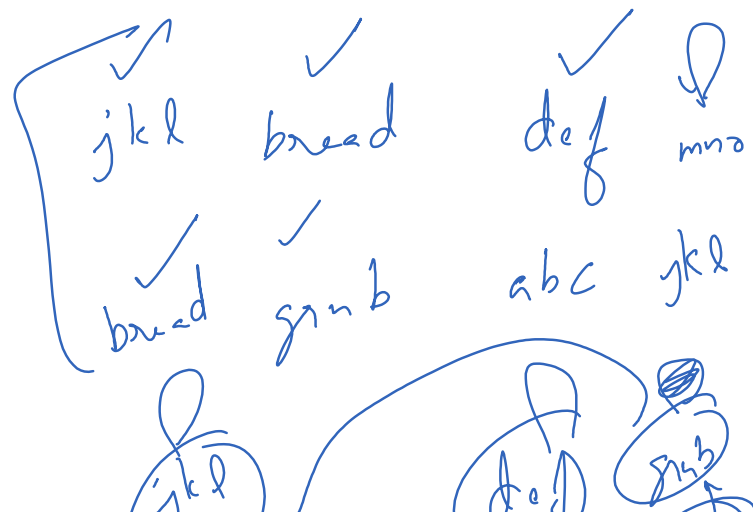
treat fight miss
like train lost

- 1
- 3
- treat fight
- miss
- like train
- lost
- 4
- treat catch
- train fight
- like catch
- miss lost

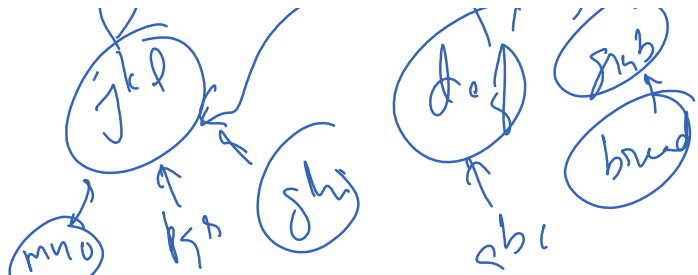
From
<https://www.pepcoding.com/resources/data-structures-and-algorithms-in-java-levelup/graphs/sentence-similarity-official/question>



- ✓ p r n → B j k l
- ✓ m n o → p r n
- ✓ a b c ↔ d e f
- ✓ g h i ↔ j k l



✓
 ✓ gmb → bread
 ✓ bread - jkl



11:45 - 11:55

abc ↔ def ✓

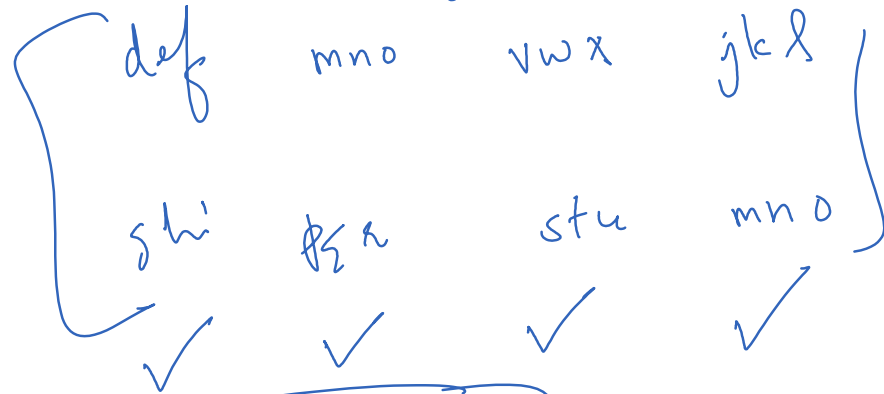
def → shi ✓

jkl - mno ✓

mno - pqr ✓

stu → vwx ✓

vwx → yz ✓



11:48 - 12:00